2. Linux Security Basics

🐧 Linux & System Security

🧠 1. What is Linux?

Linux is a **free and open-source UNIX-like operating system kernel** originally created by **Linus Torvalds** in 1991.

It forms the core of various **distributions (distros)** like Ubuntu, Kali, CentOS, Arch, etc., which package the kernel with GNU tools and services.

💡 **Why it's important for cybersecurity:**

- Used in **servers**, **embedded devices**, **routers**, **IoT**, and **infosec tools**.
- Powers platforms like **Kali Linux**, the de facto OS for **penetration testing**.

💬 2. What is a Linux Command?

A **Linux command** is a text-based instruction given to the system via a **shell** (like `bash`, `zsh`, etc.).

Examples:

- `ls` – list directory contents
- `cd` – change directory
- `chmod` – change file permissions
- `grep` – search for patterns in files
- `ps` – view running processes

📁 Commands follow the structure:

```
command [options] [arguments]
```

Commands can be:

- **Built-in** (e.g., `cd`, `echo`)
- **External binaries** (e.g., `/bin/ls`, `/usr/bin/ssh`)

🏗️ 3. What is the Structure of the Linux Operating System?

Linux OS is divided into the following **layers**:

Layer	Description
Kernel	Core of the OS; manages hardware, memory, processes, and I/O.
Shell	Interface for users to interact with the OS (e.g., `bash`, `zsh`).
Utilities	System tools (e.g., `cp`, `mv`, `apt`).
File System	Hierarchical structure defined by **FHS**.
User Programs	Apps like browsers, editors, terminals.

🛠️ For OSCP, understanding **kernel space vs. user space**, and how **system calls** bridge them, is crucial.

📂 4. What is the Purpose of the FHS (Filesystem Hierarchy Standard) and its Benefits?

FHS defines the **directory structure and directory contents** in Unix/Linux.

📌 Purpose:

- Ensures **consistency** across distributions.
- Helps developers know **where to place files**.
- Makes scripts and software **portable**.
- Simplifies **automation, backups, monitoring**.

📁 5. What Are the Different Directories in the Linux File System and Their Purposes?

Directory	Purpose
`/`	Root of the entire filesystem.
`/bin`	Essential binaries for all users (e.g., `ls`, `cp`).
`/sbin`	System binaries (used for booting, repairing).
`/etc`	System configuration files.
`/home`	User directories (`/home/karli`).
`/root`	Home directory of the `root` user.
`/var`	Variable data (logs, mail, spool files).
`/tmp`	Temporary files (cleared on reboot).
`/usr`	User-installed software and libraries.
`/lib`, `/lib64`	Libraries needed by binaries.

Directory	Purpose
`/dev`	Device files (e.g., `/dev/sda1`, `/dev/null`).
`/proc`	Virtual filesystem for process and system info.
`/boot`	Files needed for booting (e.g., GRUB, kernel).

🔒 6. How to Protect Files and Directories?

🔐 Permissions:

Use `chmod`, `chown`, `chgrp` to manage access.

Symbol	Meaning
`r`	Read
`w`	Write
`x`	Execute

Command:

```
chmod 700 secret.txt   # Owner can read/write/execute; no access to others
```

🛡️ Other techniques:

- **Use ACLs** (`setfacl`) for fine-grained permissions.
- **Immutable files**: `chattr +i filename` (can't be changed/deleted even by root).
- **Audit access** with `auditd`.
- Secure backups and restrict physical access.

🕵️ 7. How to Monitor and Investigate System Activity?

🔍 Common Tools:

Tool	Purpose
`top`, `htop`	Real-time process and memory usage.
`ps aux`	List active processes.
`lsof`	List open files and sockets.
`netstat`, `ss`	Show network connections.
`journalctl`	System logs from `systemd`.
`dmesg`	Kernel ring buffer (hardware logs).
`auditctl`, `ausearch`	Monitor security events.

🧠 **Tip:**

For forensics, focus on:

- Suspicious users (`/etc/passwd`, `who`, `last`)
- Unexpected services (`ps`, `systemctl`, `chkconfig`)
- File modifications (`stat`, `find -mtime`, `inotifywait`)
- Log tampering

🔐 8. How to Securely Transfer Files and Data?

🔒 Secure Methods:

Tool	Use Case
`scp`	Secure file copy over SSH.
`sftp`	Secure FTP-like interface using SSH.
`rsync -e ssh`	Efficient sync over SSH.
`gpg`	Encrypt files before transfer.
`curl -k --cert`	Transfer via HTTPS with certs.

Example:

```
scp file.txt karli@192.168.1.10:/home/karli/
```

💡 Use strong keys instead of passwords and verify host fingerprints.

🔥 9. How to Configure and Manage a Firewall?

Linux uses **iptables** or **nftables** (newer), and distros often include frontends like `ufw` or `firewalld`.

🔥 Tools:

Tool	Description
`iptables`	Packet filtering framework (netfilter).
`nft`	Modern replacement for `iptables`.
`ufw`	Uncomplicated Firewall (Ubuntu-friendly).
`firewalld`	Dynamic firewall daemon (RHEL/CentOS).

Example with `ufw`:

```
sudo ufw enable
sudo ufw default deny incoming
sudo ufw allow ssh
sudo ufw status
```

Example with `iptables`:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -P INPUT DROP
```

🛡️ Always **allow SSH before dropping INPUT** or risk locking yourself out!

💀 10. How to Identify and Terminate Malicious Processes?

🧭 Steps:

1. **List Processes**:

   ```
   ps aux | grep suspicious
   ```

2. **Check Resource Usage**:

   ```
   top, htop
   ```

3. **Identify Network Activity**:

   ```
   lsof -i, netstat -tunlp, ss -lntp
   ```

4. **Find Executables and Paths**:

   ```
   which processname
   ```

5. **Kill the Process**:

   ```
   kill -9 <PID>
   ```

6. **Investigate Further**:

   - Check binaries with `strings`, `file`, `md5sum`.
   - Run in sandbox (e.g., Cuckoo, strace, gdb).
   - Use `chkrootkit` or `rkhunter`.

⚠️ Don't just kill — **investigate first** to avoid destroying evidence!
